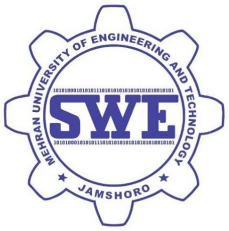


Mehran University of Engineering and Technology, Jamshoro.

Academic Year 2022-2023

Assignment Title: Complex Engineering Report

Name of Assignment: Ride-Sharing Platform

Group members Roll No.: 39, 77, 83, 125, 133, 135

Section:“I ”

Subject: Software Quality Engineering

Date of Submission:02/10/2023

Table of Contents

RIDE SHARING PLATFORM

Ride Sharing Platform

Introduction:

In today's fast-paced world, the need for efficient and convenient transportation solutions has never been greater. The "Development of a Ride-Sharing Platform" project aims to address this need by creating a robust and user-friendly platform that offers ride booking, carpooling, and secure payment processing. This project holds significant importance in improving the way people commute, reducing traffic congestion, and promoting eco-friendly transportation options.

Project Overview

Scope:

- Offer users the ability to book electric vehicles (EVs) for rides.
- Encourage carpooling as a means to reduce traffic congestion and carbon emissions.

Objectives:

- Develop a functional prototype of the ride-sharing platform.
- Implement essential functionalities including ride booking, carpooling, and payment processing.
- Create both web and mobile applications to cater to a wider user base.
- Follow the Agile Software Development Life Cycle (SDLC) process for iterative and collaborative development.

Project Requirements

The functional and non-functional requirements for the ride-sharing platform are as follows.

Functional Requirements:

1. User Registration:

- Users must be able to create accounts using email, phone number, or social media profiles.
- The system should verify user information for authenticity and security.
- Users should have the option to provide personal information and a profile picture.
- Password reset and account recovery functionality should be available.
- Users should be able to specify their role (passenger or driver) during registration.

2. Ride Booking:

- Passengers should be able to search for available rides based on their current location and destination.
- Passengers should see a list of available drivers, along with driver ratings and vehicle - information.
- Passengers should be able to request a ride, specifying the number of passengers and any special requirements.
- Drivers should receive ride requests and have the option to accept or decline them.

Upon acceptance, passengers should receive confirmation with estimated arrival time and driver details.

- Passengers and drivers should be able to communicate through in-app messaging or calls.
- Users must be able to specifically book electric vehicles (EVs) for rides to reduce carbon emissions.

3. Carpooling:

- The platform must support carpooling where multiple passengers can share a ride.
- Passengers should be able to opt for carpooling during ride booking.

4. Payment Processing:

- Users should be able to link their payment methods (credit card, PayPal, etc.) securely.
- The platform should calculate and display fare estimates before confirming the ride.
- Payments should be processed automatically after the completion of the ride.
- Users should receive digital receipts for their rides.

Non Functional Requirements:

1. Performance:

- The platform must handle a large number of concurrent users and transactions efficiently.
- Response times for ride requests and updates should be minimal.

2. Security:

- Authentication and authorization mechanisms should be robust to prevent unauthorized access.
- Response times for ride requests and updates should be minimal.

3. Sacalability:

- The system should be designed to scale horizontally to accommodate increasing user and driver numbers.
- The platform should provide a mechanism for passengers and drivers to rate and review each other.
-

4. Feedback and Rating system:

- The platform should provide a mechanism for passengers and drivers to rate and review each other.

System Architecture and Design

UseCase diagram

To summarize the details of Ride Sharing system's users (also known as actors) and their interactions with the system.

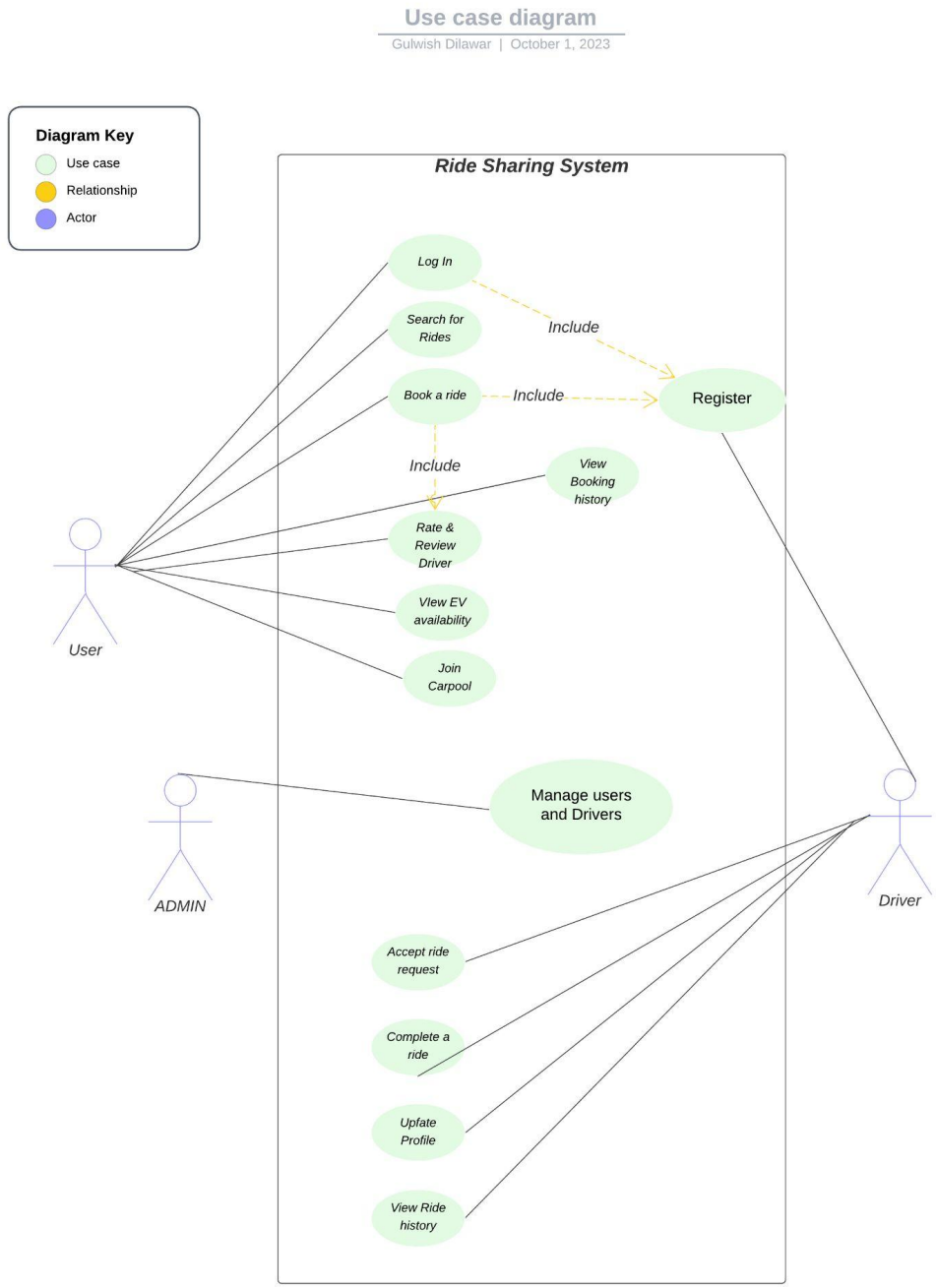


Fig 5.1.1

Class diagram

Representing the core entities and objects in the ride-sharing platform.

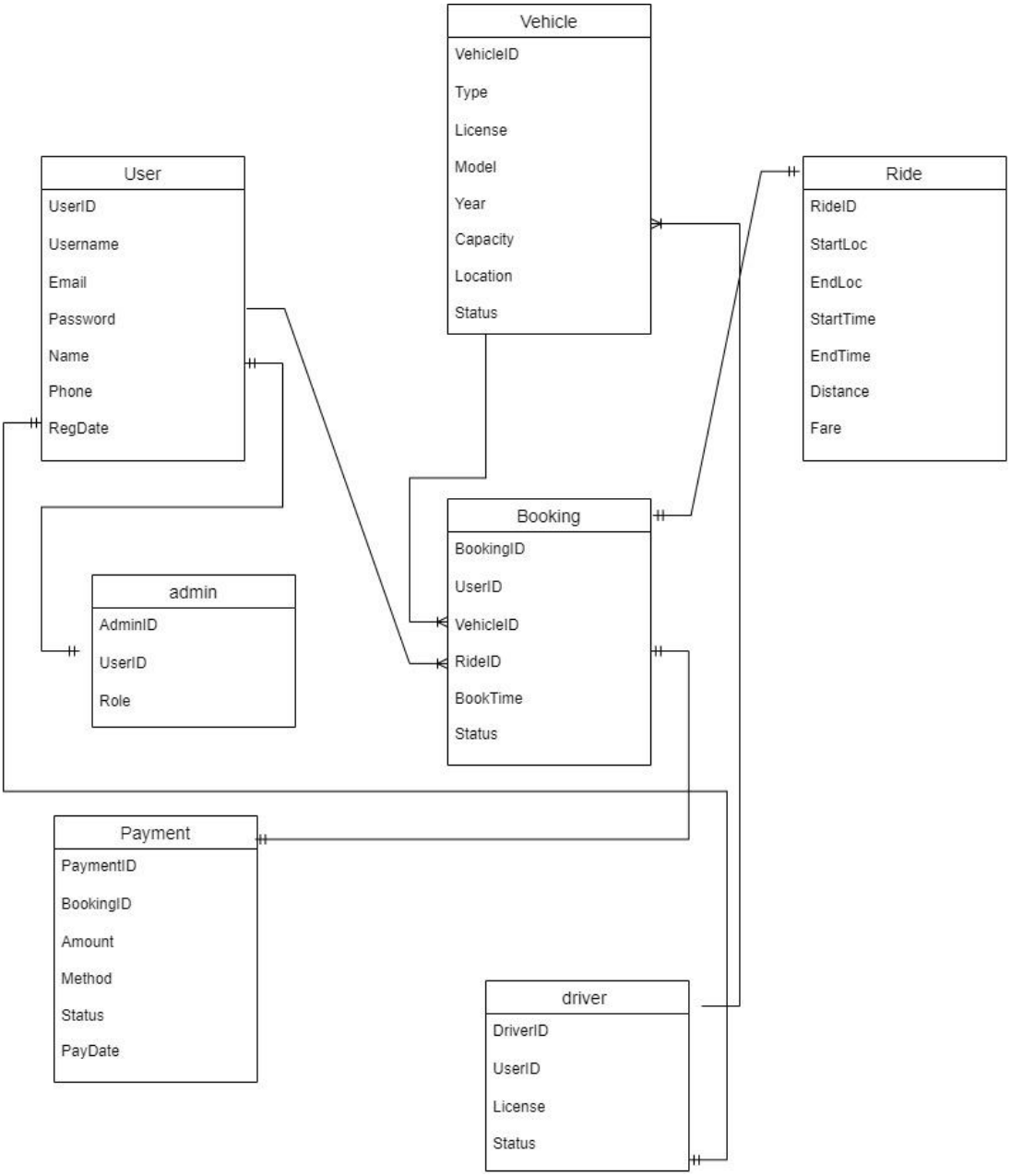


Fig 5.1.2

Entities:

1. The "User" entity stores information about registered users.
2. The "Vehicle" entity contains details about each vehicle in the system.
3. The "Ride" entity records information about each individual ride.
4. The "Booking" entity links users to their booked rides.
5. The "Payment" entity stores payment details associated with each booking.
6. The “Driver” entity stores information about registered drivers.
7. The “Admin” entity stores the information about the user as well as admin.

Relationships:

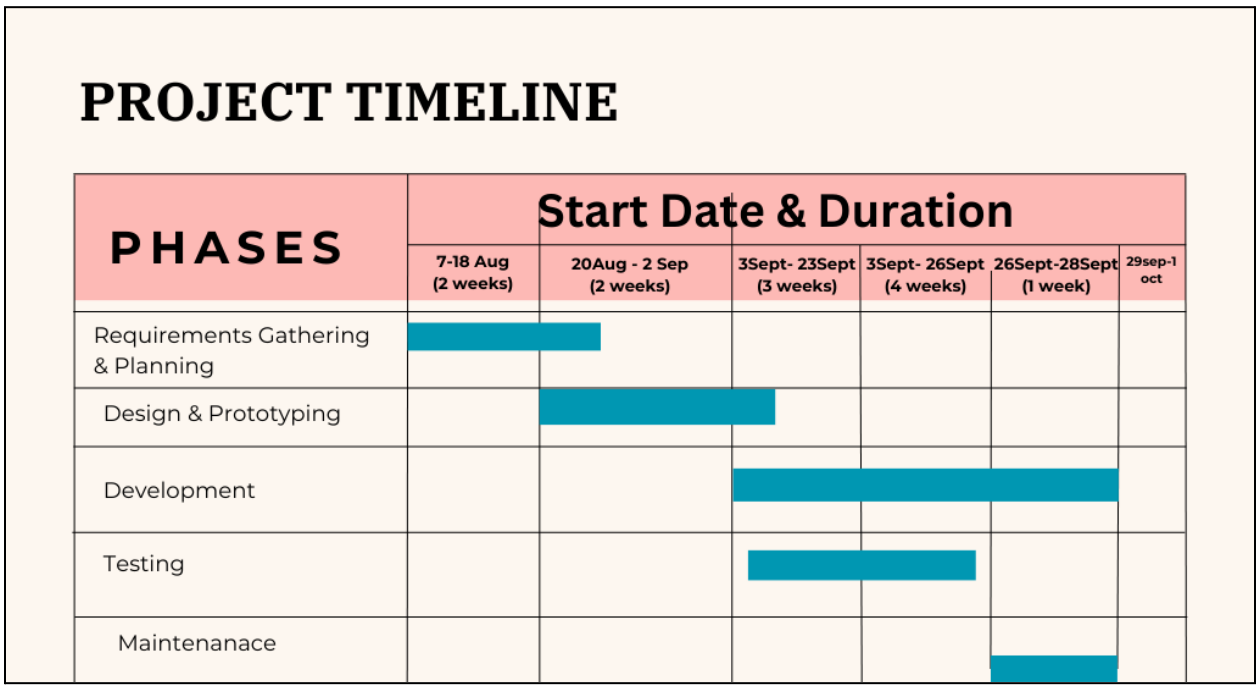
1. One User can have many Bookings (One-to-Many):
   - User (UserID) -> Booking (UserID)

2. One Vehicle can be part of many Bookings (One-to-Many):
   - Vehicle (VehicleID) -> Booking (VehicleID)

3. One Booking corresponds to one Ride (One-to-One):
   - Booking (RideID) -> Ride (RideID)

4. One Booking can have one Payment (One-to-One):
   - Booking (BookingID) -> Payment (BookingID)

Project Timeline and Effort



1. Planning & Analysis

gathering business requirements from project scope. Evaluated the cost of developomnet effort and scope of project.

2. Define Requirements

converting the information gathered during the planning and analysis phase into clear requirements for the development team. This process guides the development of several important documents: a software requirement specification (SRS), a Use Case document, and a Requirement Traceability Matrix document.

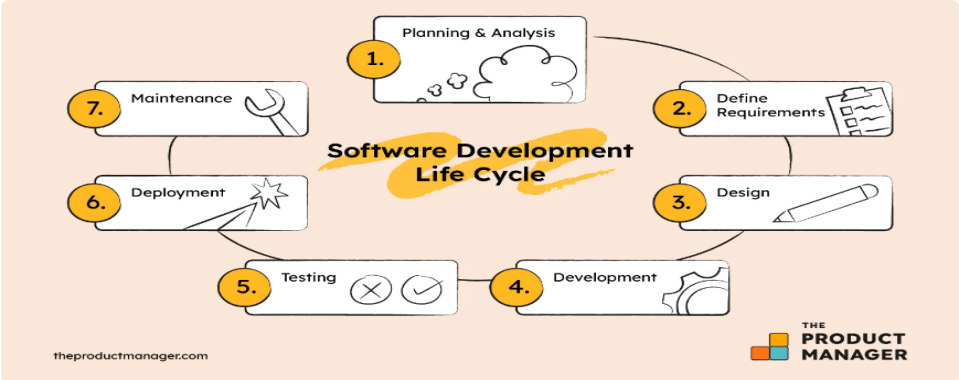
3. Design

The original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures.

To visualize what the product will look like and make changes without having to go through the hassle of rewriting code.

4. Development

the development team members divided the project into software modules and turn the software



requirement into code that makes the product.

In some cases, the development stage can also merge with the testing stage where certain tests are run to ensure there are no critical bugs.

5. Testing

Before getting the software product out the door to the production environment, it's important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it's meant to do. The testing process can also help hash out any major user experience issues and security issues.

In some cases, software testing can be done in a simulated environment. Other simpler tests can also be automated.

The types of testing to do in this phase:

- Performance testing: Assesses the software's speed and scalability under different conditions
- Functional testing: Verifies that the software meets the requirements
- Unit-testing: Tests individual units or components of the software
- Usability testing: Evaluates the software's user interface and overall user experience
- Acceptance testing: Also termed end-user testing, beta testing, application testing, or field testing, this is the final testing stage to test if the software product delivers on what it promises

6. Deployment(Alpha Release)Alpha Version

- the software that is in its first phase of software testing.to ensure that all modules are integrated properly and functioning as expected.
- Alpha testing is performed by internal employees or developers.
- This type of test is performed at the developer's site but not at the customer's site. Alpha testing is conducted after the completion of the system testing and before the beta testing. This test is done to find out bugs or defects related to usability, functionality, and consistency.

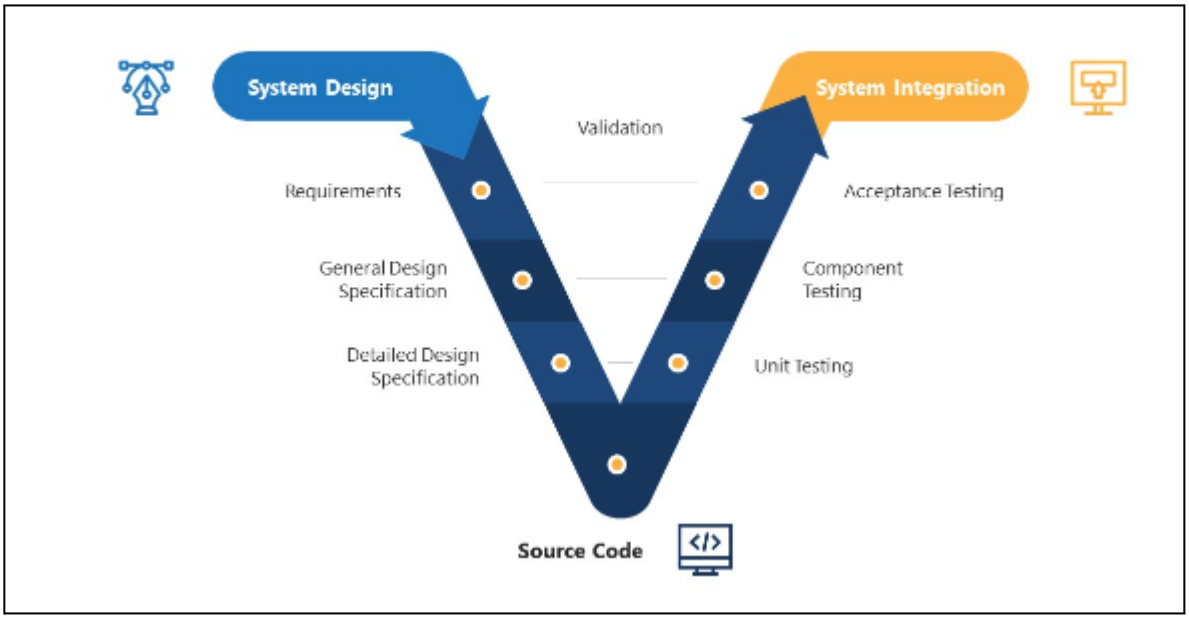
7. Maintenance

the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

In the maintenance stage, users may find bugs and errors that were missed in the earlier testing phase. These bugs need to be fixed for better user experience and retention. In some cases, these can lead to going back to the first step of the software development life cycle.

Software Development LifeCycle

Methodology



Quality Assurance process

Quality assurance is a critical aspect of ensuring the reliability and effectiveness of the eco-friendly ride-sharing platform.

We Performed quality assurance activities to design a comprehensive set of test cases covering user scenarios, vehicle availability, payment processing, and navigation.

Along with Present graphical representations of test cases executed at different SDLC lifecycle stages, showing passed and failed cases with explanations.

Test Cases

Planning and Designing

Ride Booking:

Test ride booking with valid locations and preferences:

- Preconditions: User is logged in and on the ride booking screen.
- Input: Provide valid pickup and drop-off locations, select ride preferences (e.g., vehicle type, carpooling).
- Expected Outcome: The system should find available drivers, calculate the fare, and display ride confirmation details.

Test ride booking with invalid or unreachable locations:

- Preconditions: User is logged in and on the ride booking screen.
- Input: Provide invalid or unreachable pickup and drop-off locations.
- Expected Outcome: The system should display an error message indicating that the locations are invalid or unreachable.

Verify that users receive ride confirmation details:

- Preconditions: User has successfully booked a ride.
- Expected Outcome: After confirming the ride, the user should receive confirmation details, including driver information, vehicle details, and estimated arrival time.

Carpooling:

Test carpooling by booking rides with multiple passengers:

- Preconditions: Users are logged in and on the ride booking screen.
- Input: Select the carpooling option and specify the number of passengers.
- Expected Outcome: The system should match multiple passengers for carpooling, calculate individual fares, and display confirmation details for each passenger.

TEST CASE TEMPLATE

Test Case #: ER001	Test Case Name: User login with valid information
System: Eco Ride application	Subsystem: User Registeration

Designed by: test designer	Design Date: Jul 31, 2023
Executed by: tester	Execution Date: Jul 31, 2023
Short Description: Verify that users can successfully register on the system with valid information and that they receive a verification code for account activation.	

Pre Condition:
- The mobile application is installed and launched on the device.
– the registration page is loaded.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Launch the registration page of the system. Pass	The registration page is displayed successfully.	Pass	
2	Enter valid registration information, including: - Valid email address - Valid phone number	The system accepts the information without errors.	Pass	
3	Submit the registration form. Pass	TThe system processes the registration request.	Pass	
4	Check the registered email inbox or SMS for a verification code.	A verification code is received within a reasonable time.	Pass	
5	Verify that the user is logged in after successful registration.	TThe user is redirected to their account or homepage after registration.	Pass	
6	Attempt to register another user with the same email address or phone number. Pass	The system should display an error message indicating that the email or phone number is already in use.	Pass	
7	Attempt to register another user with the same email address or phone number. Pass	The system should display an error message indicating that the email or phone number is already in use.	Pass	
8	Attempt to register with an invalid email format (e.g., without "@" symbol).	The system should display an error message indicating that the email format is invalid.	Pass	
	Attempt to register with a duplicate phone number.	The system should display an error message indicating that the phone number is already in use.		

Post-Conditions: -Post-Conditions: - A new user account is successfully registered. - The user is logged in and able to access their account.

This use case outlines the key steps and interactions involved in the "User Registration" scenario, including user input validation and the verification process. It can serve as a basis for designing and testing the registration functionality on the EcoRide ride-sharing platform.

Test cases at SDLC phases

1.Unit Testing :

Objective: Verify the correctness of individual components and functions within the ride-sharing platform.

- Detailed test scenarios are created to cover a wide range of Booking Functionality.
  - Confirm that the booking function processes ride requests accurately.
  - Test edge cases such as booking rides for various distances and different user profiles.

2.Integration Testing:

Objective: Ensure that different modules and components of the ride-sharing platform work together seamlessly.

Test Cases:

User-Driver Interaction:

- Verify that the user can successfully book rides with available drivers.
- Test communication between the user and driver modules.

Carpooling Integration:

- Confirm that carpooling features integrate smoothly with ride booking.
- Test scenarios where multiple users share rides.

Payment Integration:

- Ensure that payment processing integrates with booking and that payments are processed correctly.

3.System Testing Objective: Assess the entire eco-friendly ride-sharing platform as a unified system, testing functionality, performance, and user experience.

Test Cases:

End-to-End Ride Booking:

- Simulate real-world scenarios where users search, book, and complete rides from start to finish.

Performance Testing:

- Evaluate system performance under varying loads, including peak usage times.
- Assess response times, scalability, and resource utilization.

User Experience (UX) Testing:

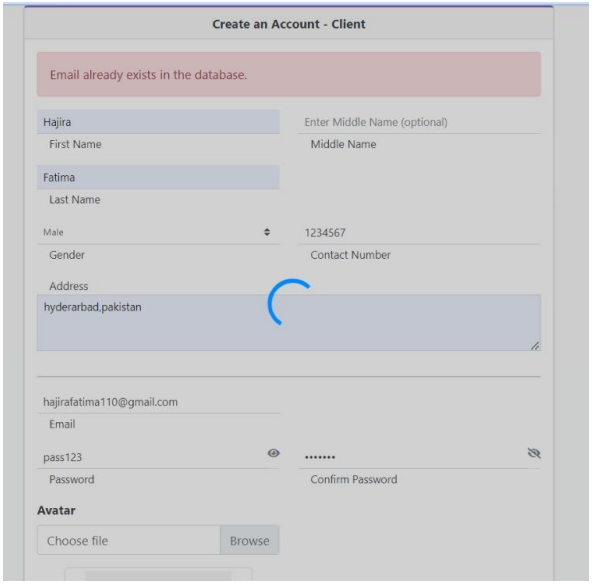
- Test the platform's user interface (UI) for consistency, responsiveness, and ease of use.
- Verify that users can navigate and interact with the platform intuitively.

Testing Tools and Environments

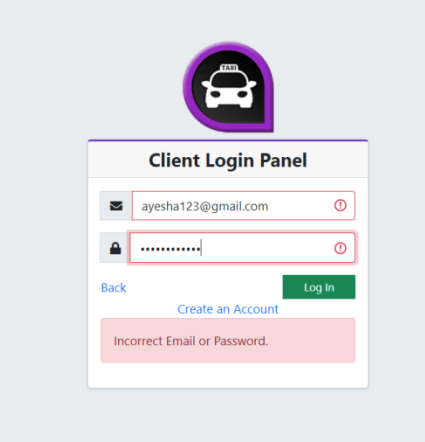
Test cases execution & Graphical Representations 7.2.1

Passed and Failed Cases

TestCase1: Duplicate Registration not allowed

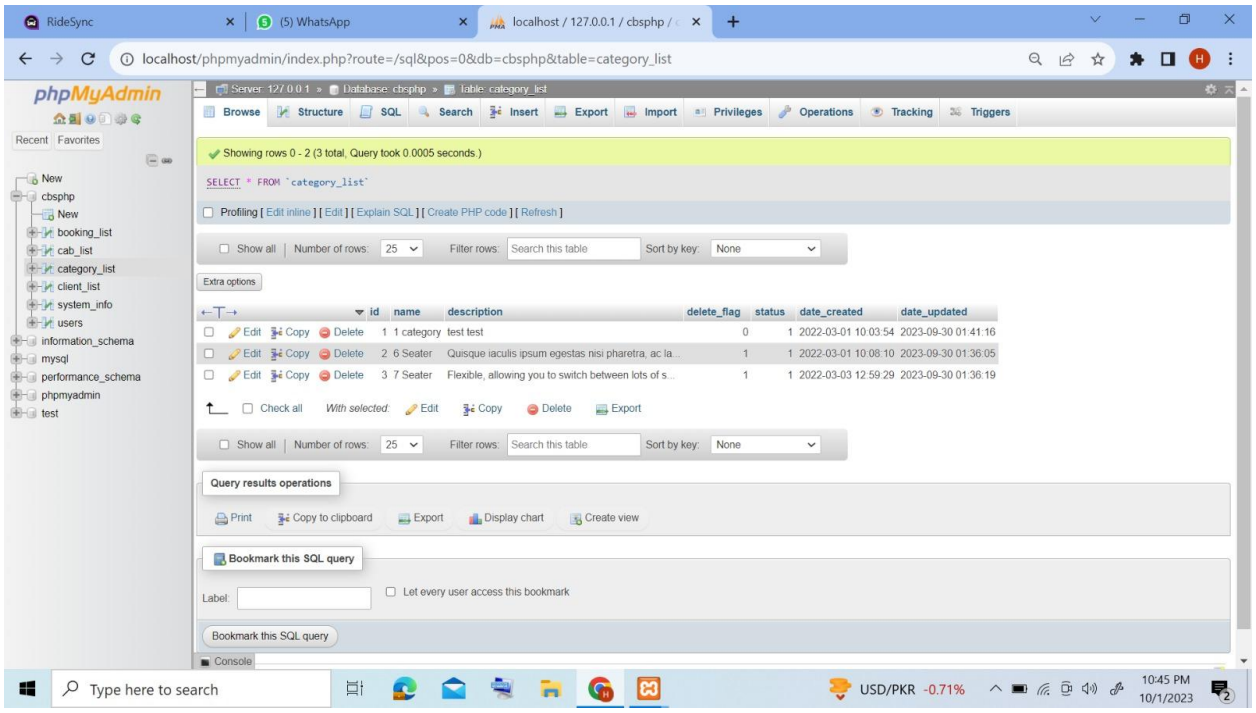


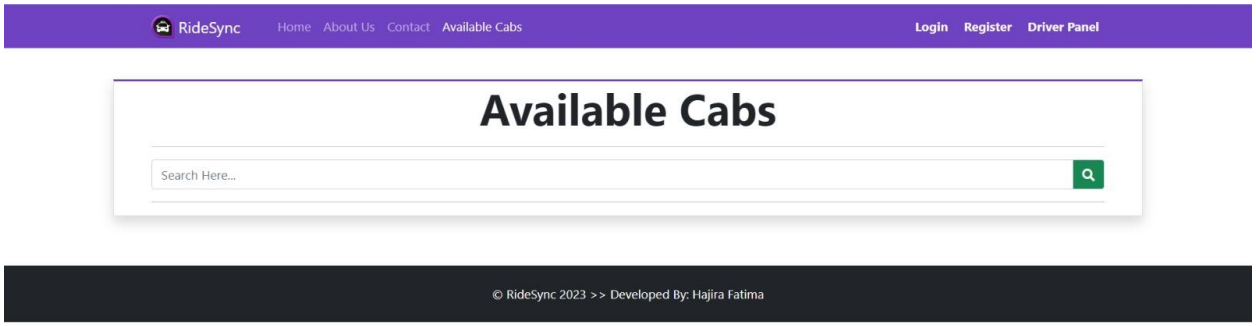
TestCase2:Incorrect Acess Credentials

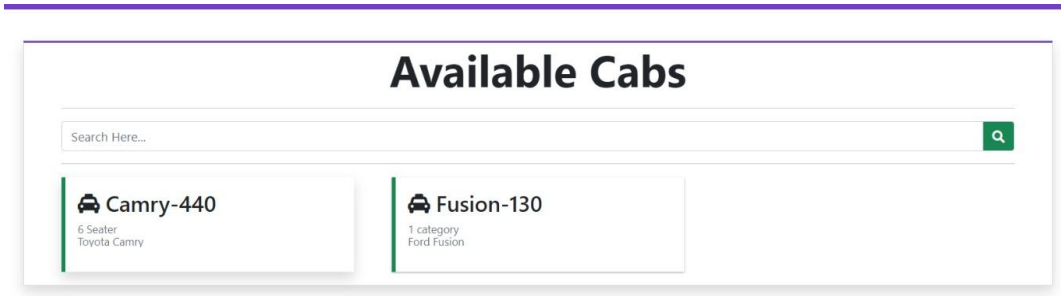


TestCase3: Available cabs not showing even though entered in DB

Until status dropedoff from driver site is not confirmed.

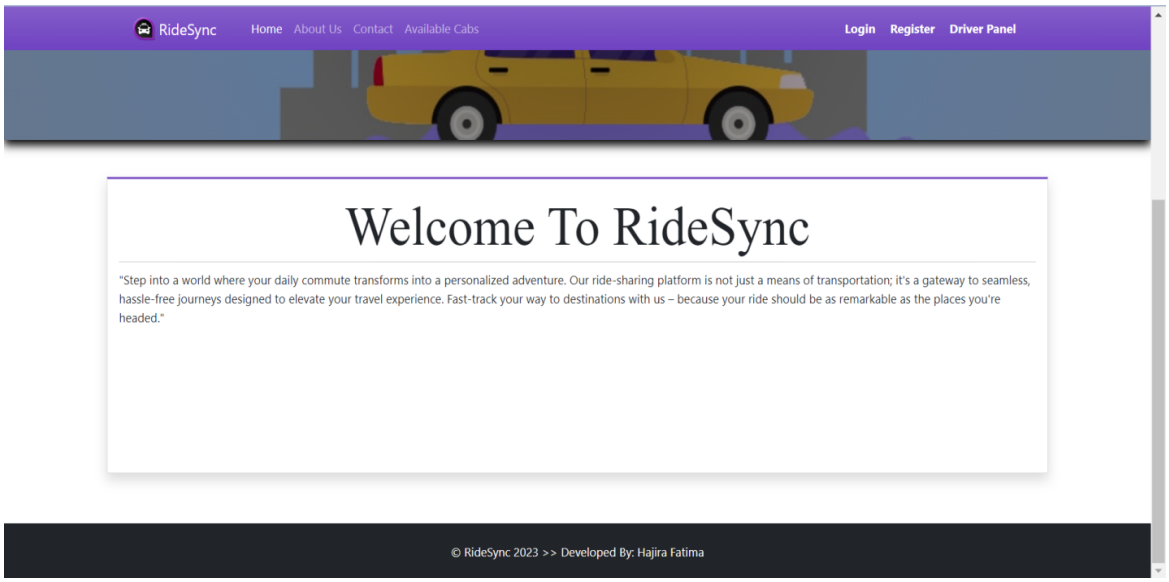

RideSync
(5) WhatsApp
localhost / 127.0.0.1 / cbsphp /
localhost/phpmyadmin/index.php?route=/sql&pos=0&db=cbsphp&table=category_list
phpMyAdmin
Recent Favorites
New
cbsphp
New
booking_list
cab_list
category_list
client_list
system_info
users
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: cbsphp » Table: category_list
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)
SELECT * FROM `category_list`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
id name description delete_flag status date_created date_updated
Edit Copy Delete 1 1 category test test 0 1 2022-03-01 10:03:54 2023-09-30 01:41:16
Edit Copy Delete 2 6 Seater Quisque iaculis ipsum egestas nisi pharetra, ac la... 1 1 2022-03-01 10:08:10 2023-09-30 01:36:05
Edit Copy Delete 3 7 Seater Flexible, allowing you to switch between lots of s... 1 1 2022-03-03 12:59:29 2023-09-30 01:36:19
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query
Console
Type here to search
USD/PKR -0.71%
10:45 PM
10/1/2023

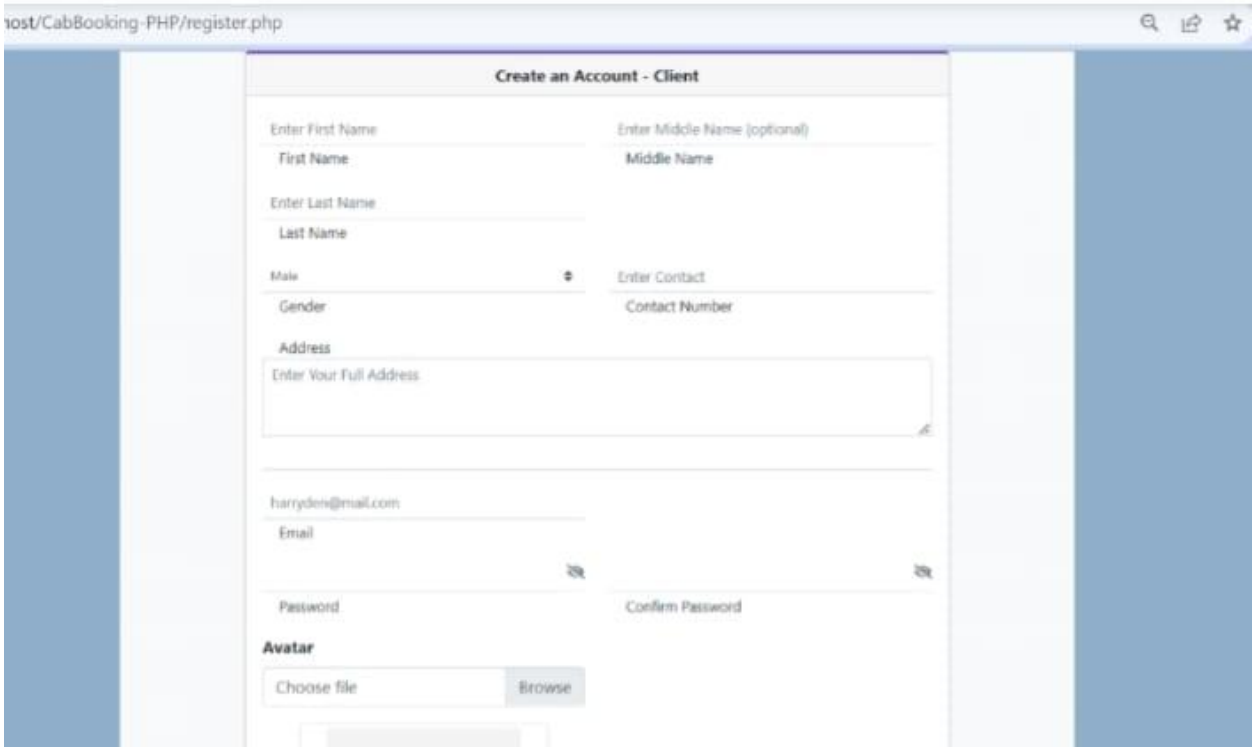

RideSync
Home About Us Contact Available Cabs
Login Register Driver Panel
Available Cabs
Search Here...
© RideSync 2023 >> Developed By: Hajira Fatima

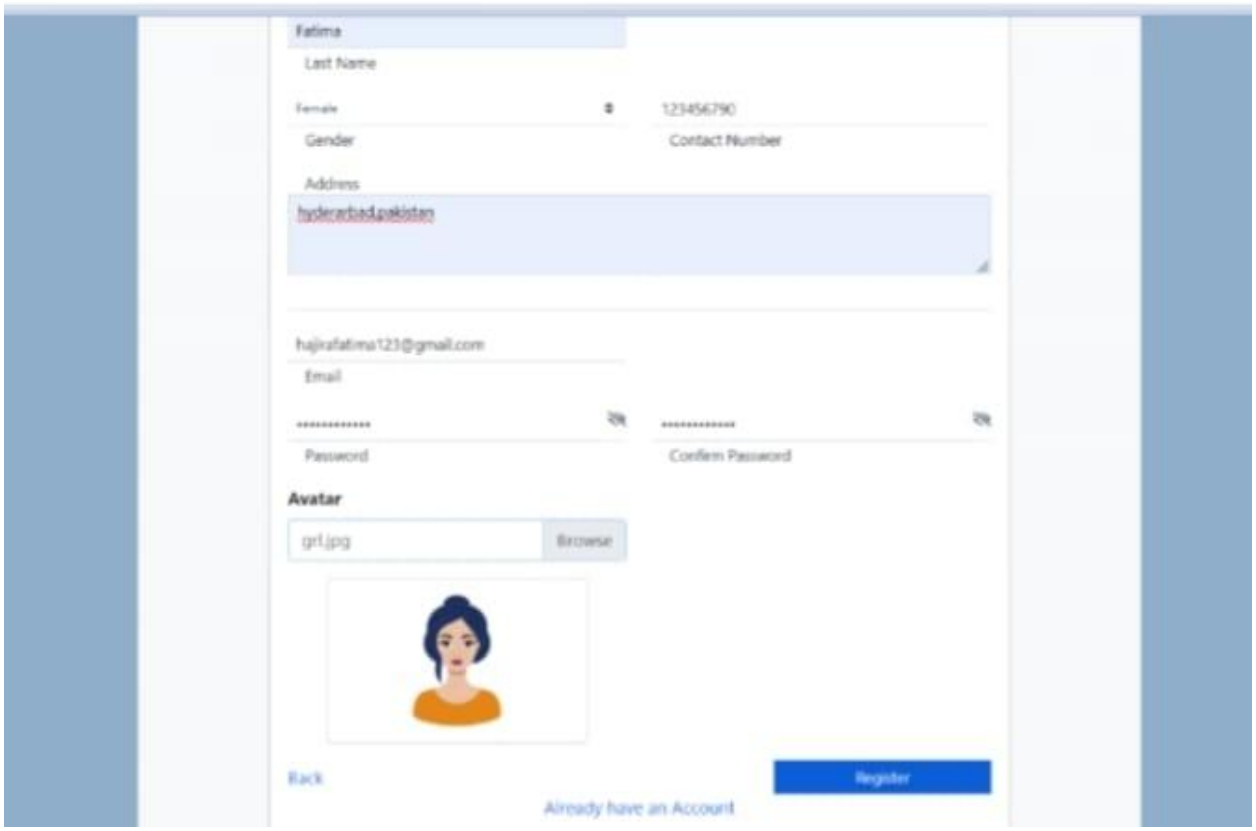


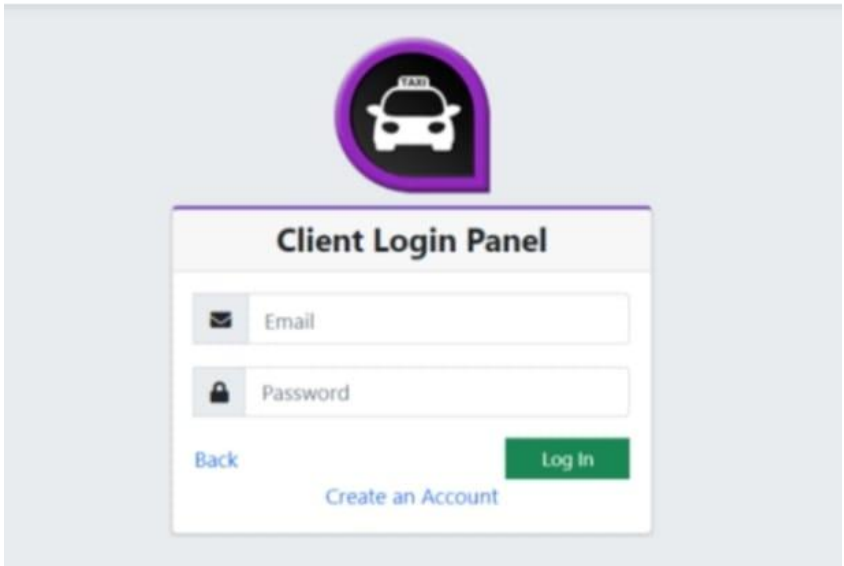
Prototype Development

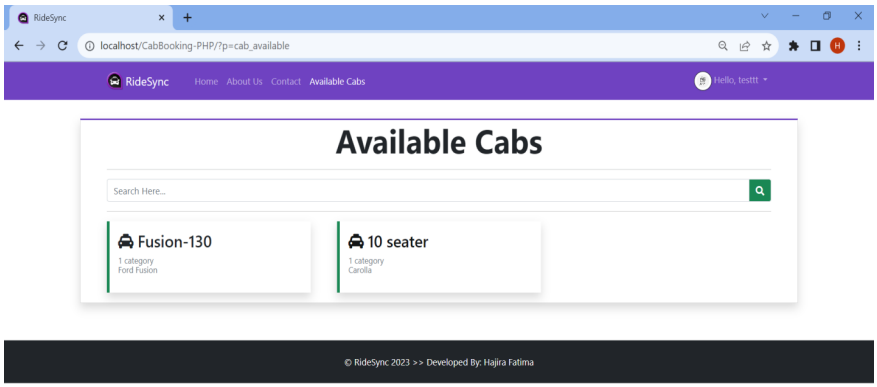
Web App:


RideSync
Home
About Us
Contact
Available Cabs
Login
Register
Driver Panel
Welcome To RideSync
"Step into a world where your daily commute transforms into a personalized adventure. Our ride-sharing platform is not just a means of transportation; it's a gateway to seamless, hassle-free journeys designed to elevate your travel experience. Fast-track your way to destinations with us – because your ride should be as remarkable as the places you're headed."
© RideSync 2023 >> Developed By: Hajira Fatima

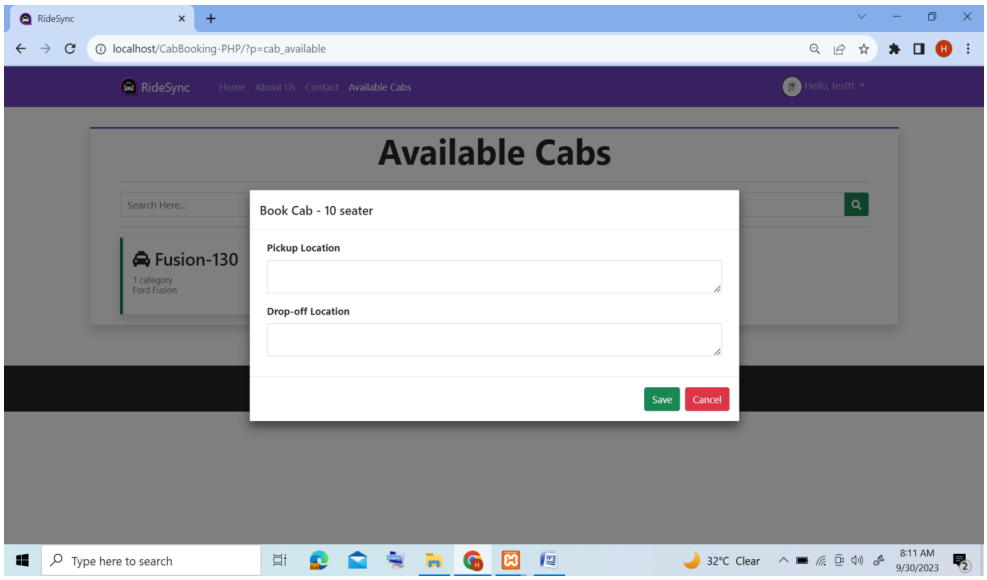

ıost/CabBooking-PHP/register.php
Create an Account - Client
Enter First Name
First Name
Enter Middle Name (optional)
Middle Name
Enter Last Name
Last Name
Male
Gender
Enter Contact
Contact Number
Address
Enter Your Full Address
harrydes@mail.com
Email
Password
Confirm Password
Avatar
Choose file
Browse


Fatima
Last Name
Female
Gender
123456790
Contact Number
Address
hyderabad,pakistan
hajirafatima123@gmail.com
Email
Password
Confirm Password
Avatar
girl.jpg
Browse
Back
Register
Already have an Account

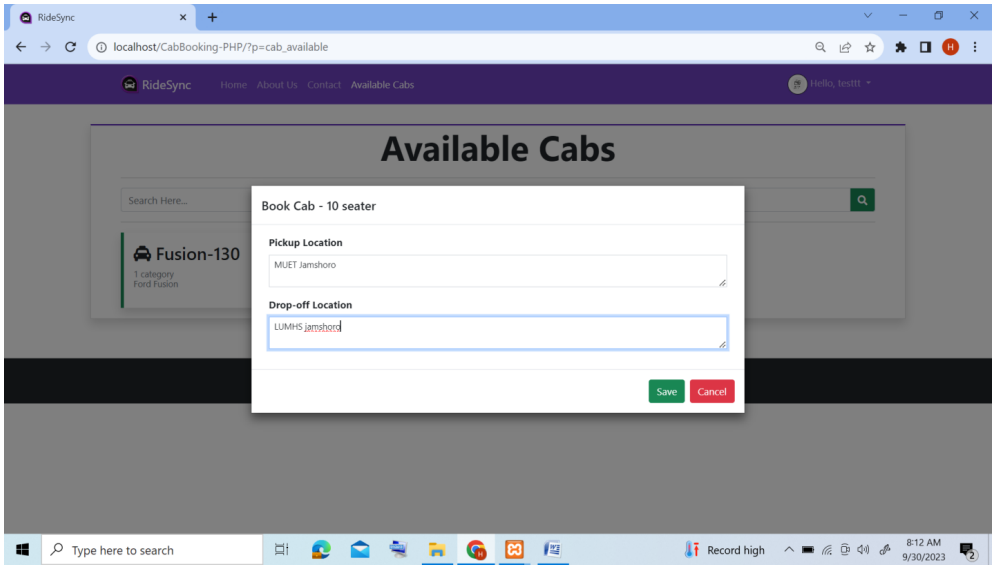

Client Login Panel
Email
Password
Back
Log In
Create an Account

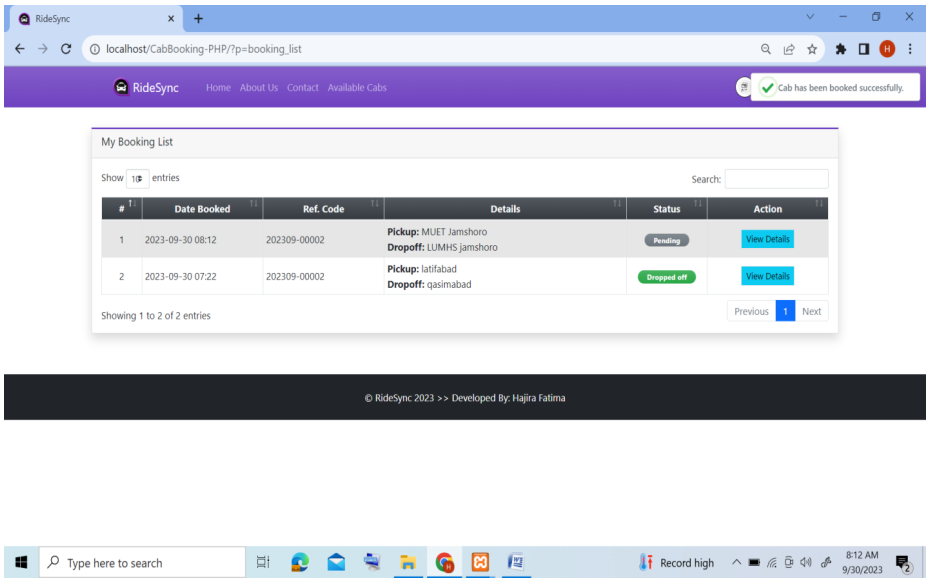


Booking ride 10 seater



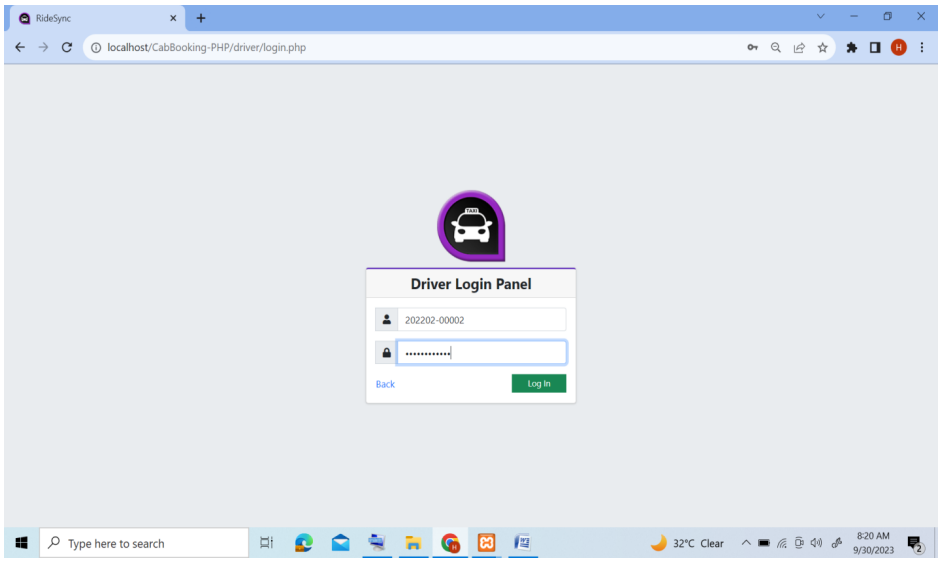
Giving pickup and drop off location

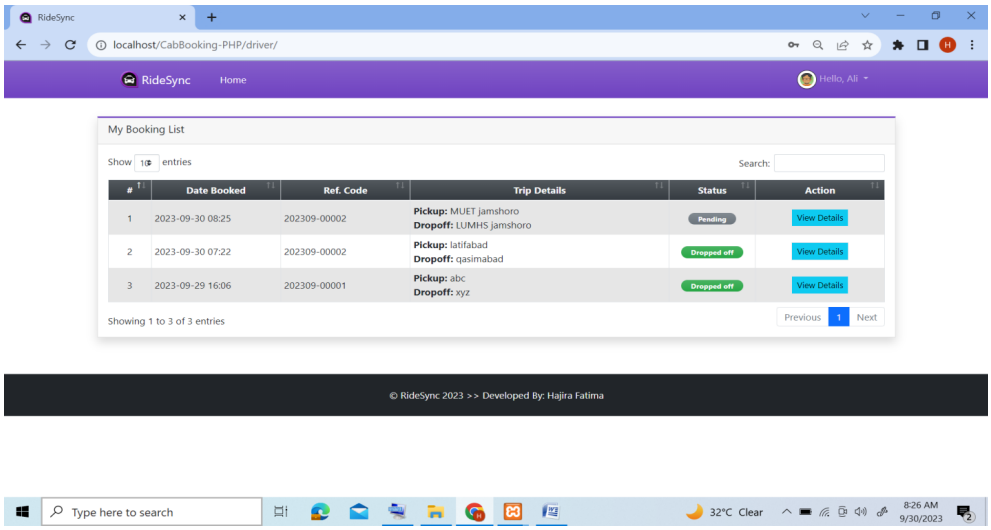




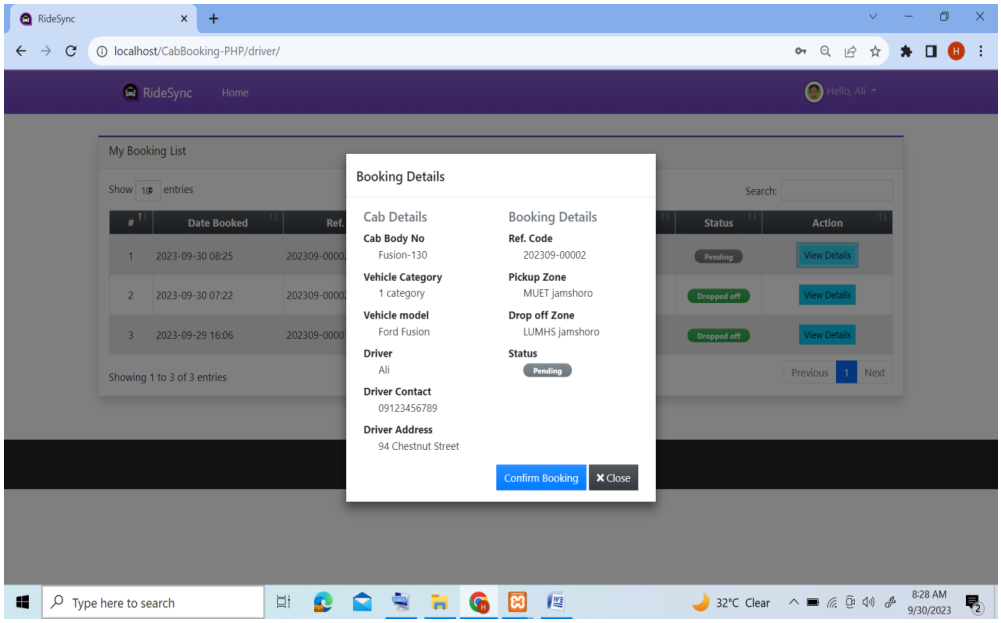
here User can veiw details of the ride which they book here the request is in pending and going towards driver

DRIVER PANEL

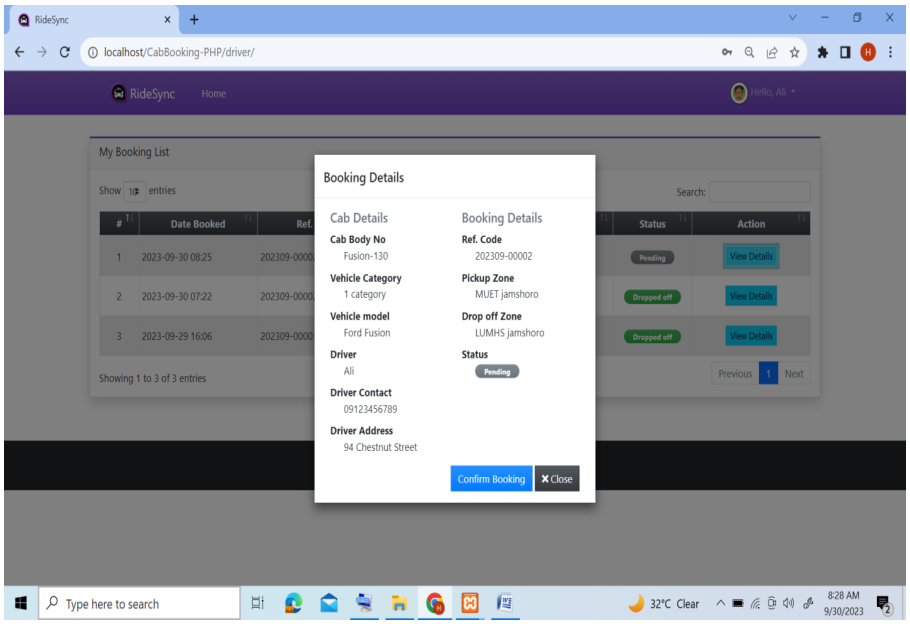




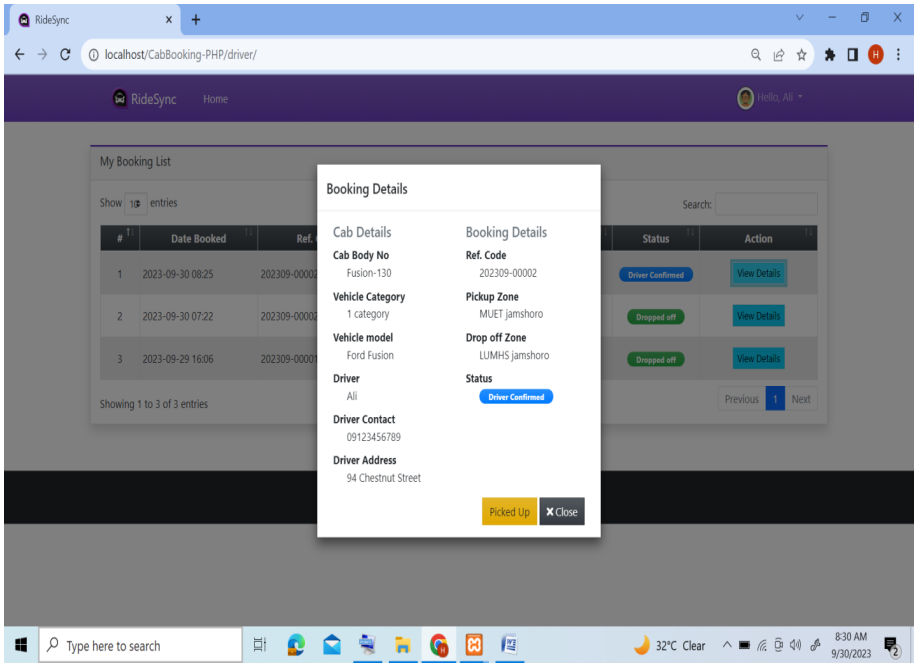
This view s for driver here he can see the request from client or user of ride which they book



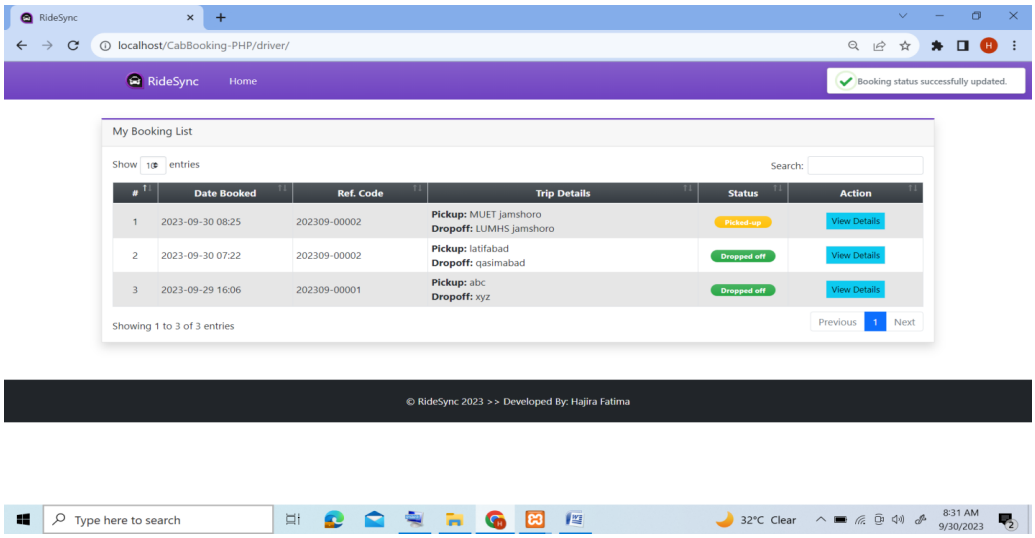
Booking confirm by driver



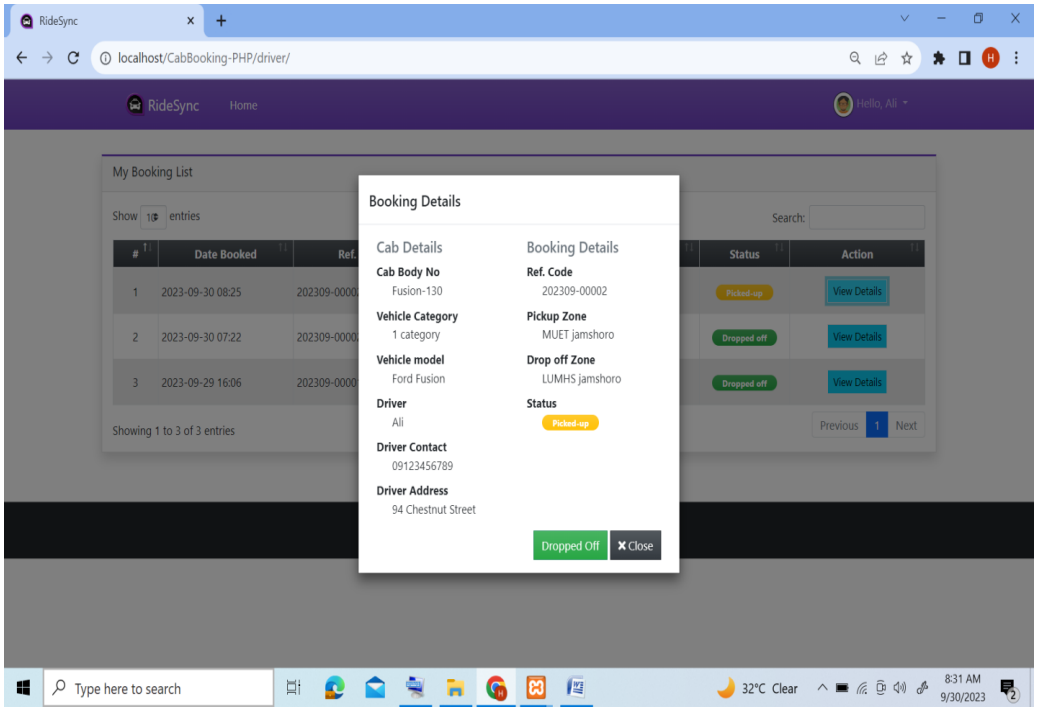
Booking confirmed



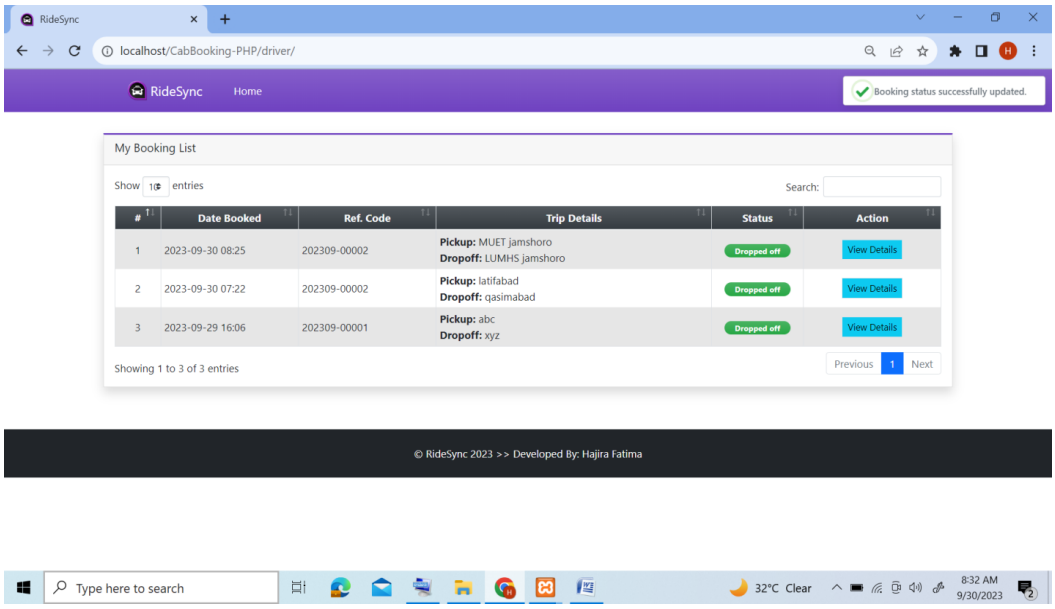
Client picked up by driver



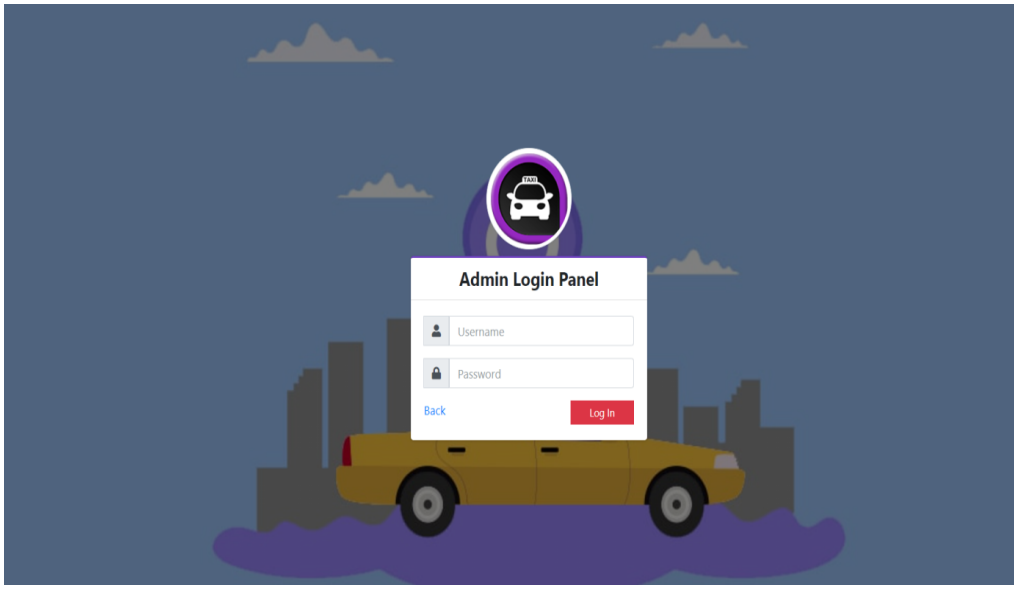
Client picked



Client drop-off by driver

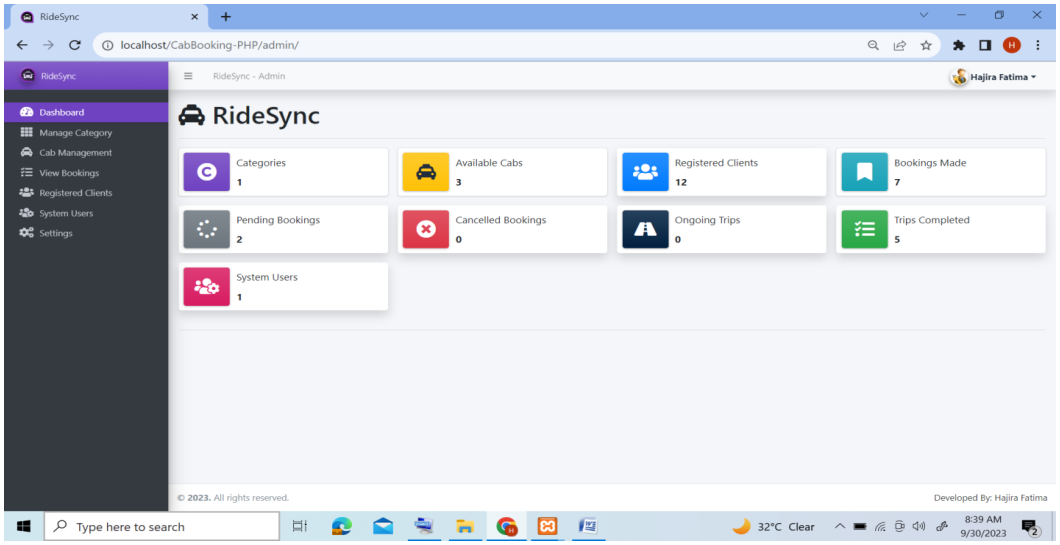


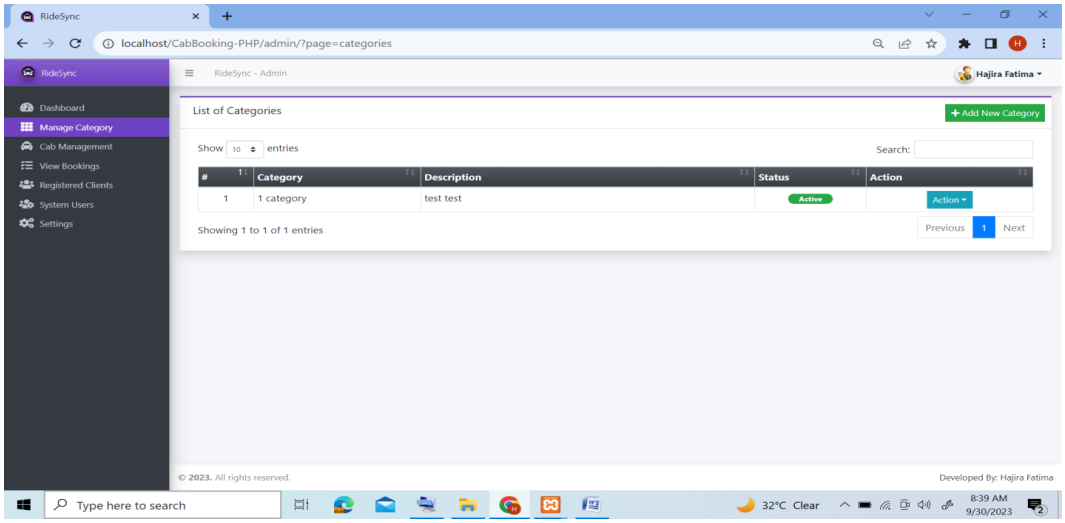
ADMIN PANEL

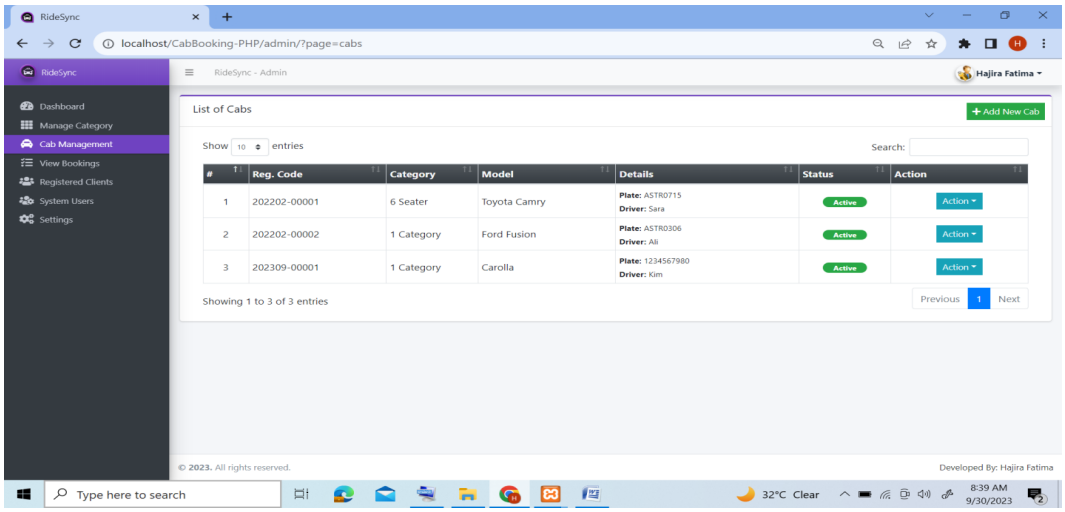


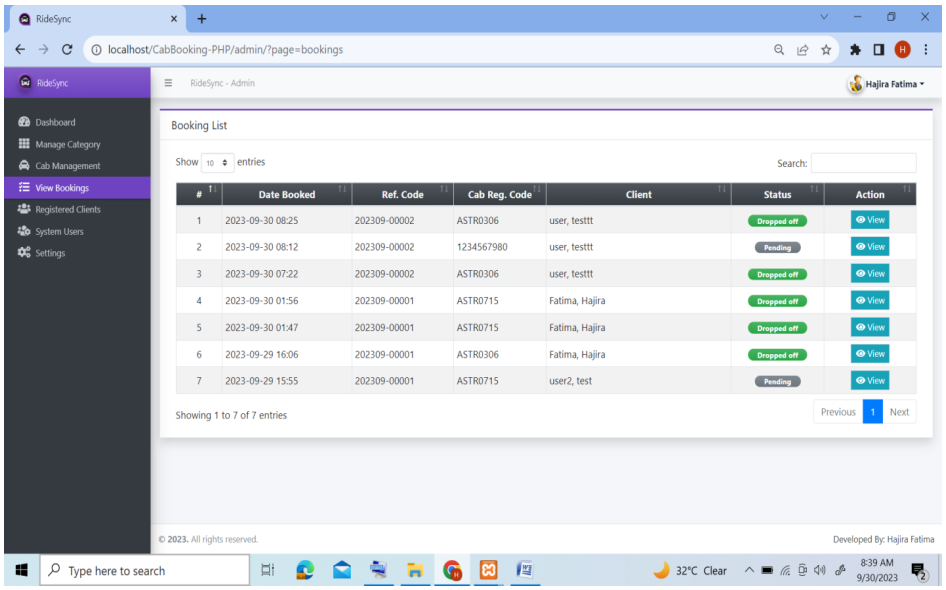
Ths view is availible to admin

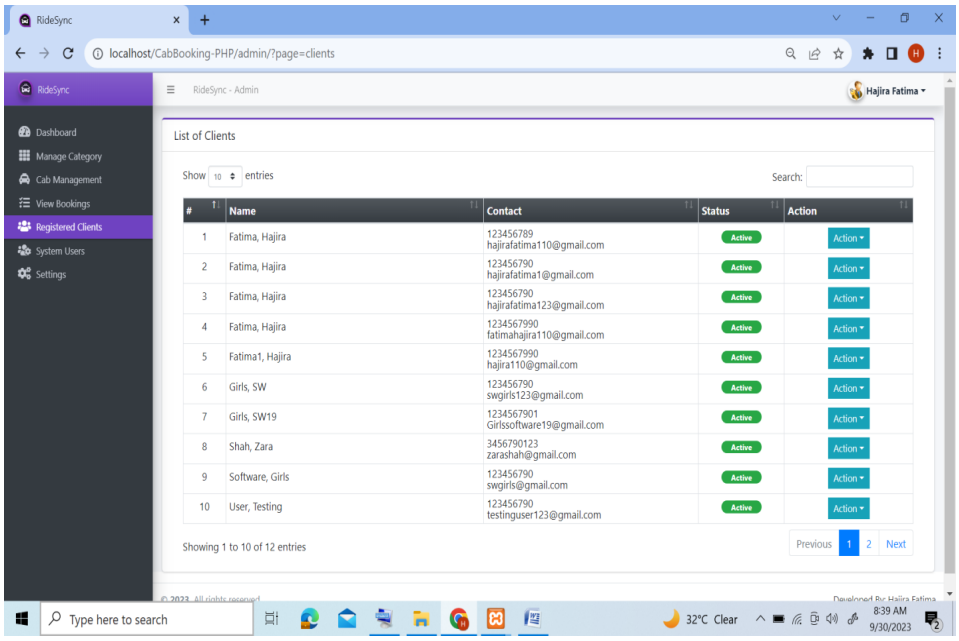
Admin can view following things

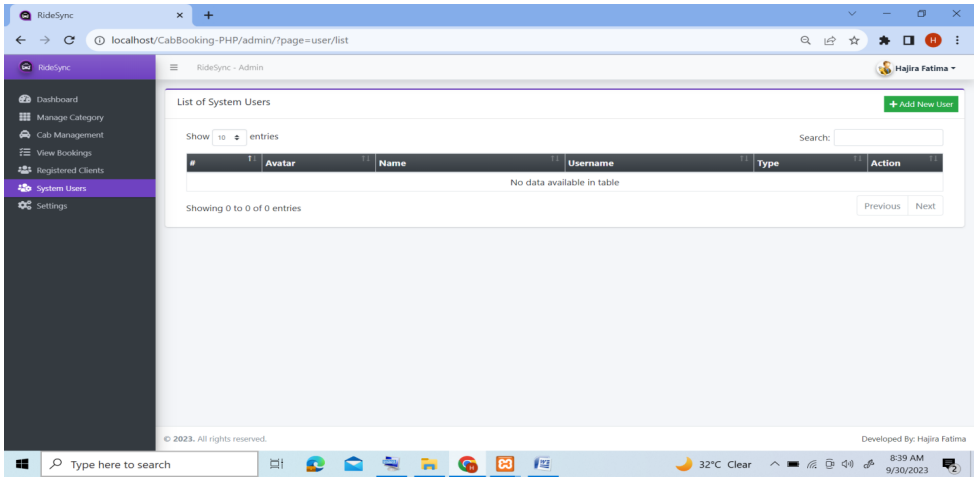




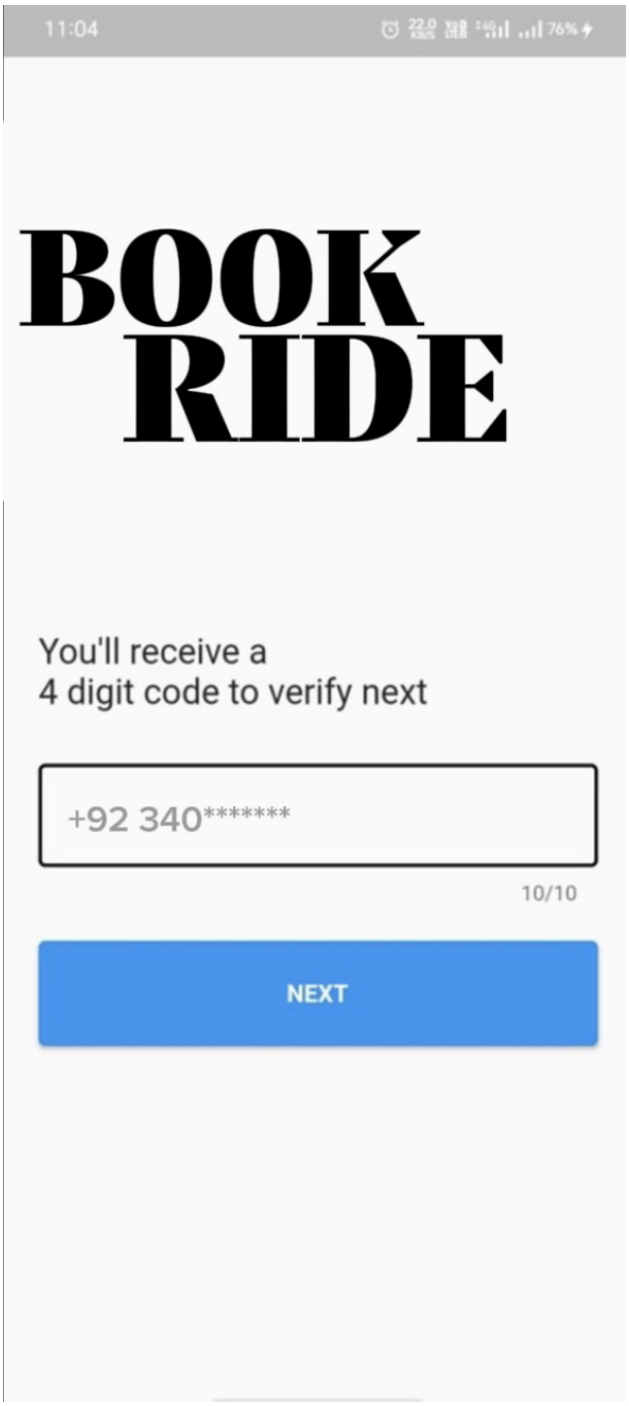

RideSync
localhost/CabBooking-PHP/admin/?page=cabs
RideSync - Admin
Hajira Fatima
Dashboard
Manage Category
Cab Management
View Bookings
Registered Clients
System Users
Settings
List of Cabs
Add New Cab
Show 10 entries
Search:
Reg. Code Category Model Details Status Action
1 202202-00001 6 Seater Toyota Camry Plate: ASTR0715 Driver: Sara Active Action
2 202202-00002 1 Category Ford Fusion Plate: ASTR0306 Driver: Ali Active Action
3 202309-00001 1 Category Carolla Plate: 1234567980 Driver: Kim Active Action
Showing 1 to 3 of 3 entries
Previous 1 Next
© 2023. All rights reserved.
Developed By: Hajira Fatima
Type here to search
32°C Clear
8:39 AM
9/30/2023

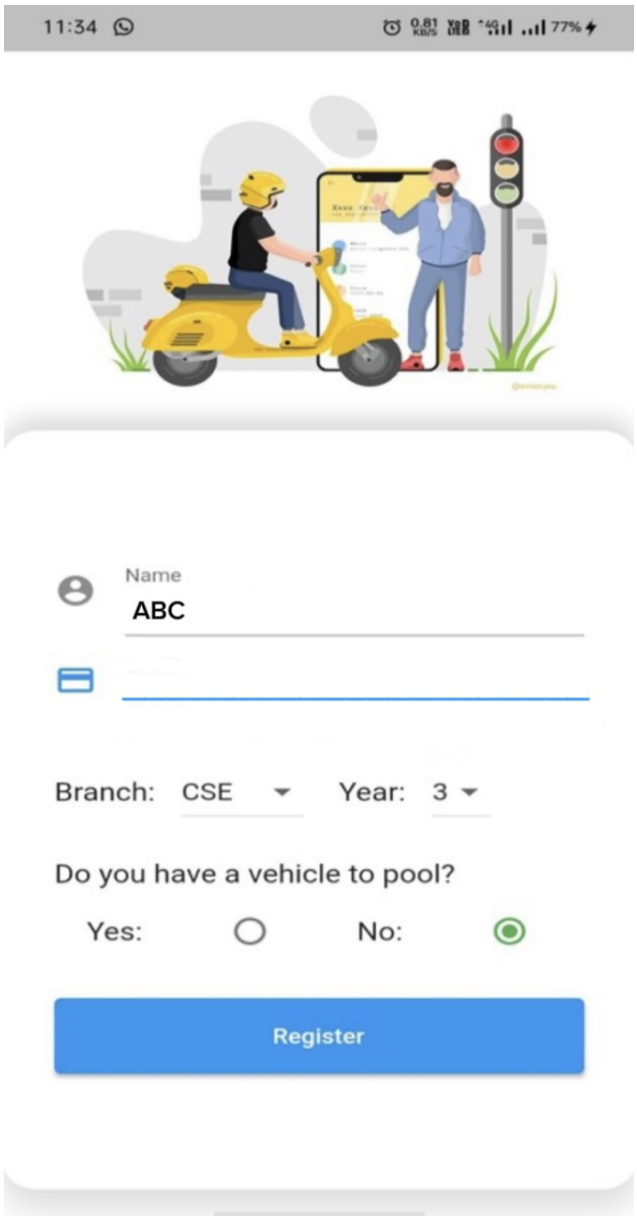

RideSync
localhost/CabBooking-PHP/admin/?page=bookings
RideSync - Admin
Hajira Fatima
Dashboard
Manage Category
Cab Management
View Bookings
Registered Clients
System Users
Settings
Booking List
Show 10 entries
Search:
Date Booked Ref. Code Cab Reg. Code Client Status Action
1 2023-09-30 08:25 202309-00002 ASTR0306 user, testtt Dropped off View
2 2023-09-30 08:12 202309-00002 1234567980 user, testtt Pending View
3 2023-09-30 07:22 202309-00002 ASTR0306 user, testtt Dropped off View
4 2023-09-30 01:56 202309-00001 ASTR0715 Fatima, Hajira Dropped off View
5 2023-09-30 01:47 202309-00001 ASTR0715 Fatima, Hajira Dropped off View
6 2023-09-29 16:06 202309-00001 ASTR0306 Fatima, Hajira Dropped off View
7 2023-09-29 15:55 202309-00001 ASTR0715 user2, test Pending View
Showing 1 to 7 of 7 entries
Previous 1 Next
© 2023. All rights reserved.
Developed By: Hajira Fatima
Type here to search
32°C Clear
8:39 AM
9/30/2023

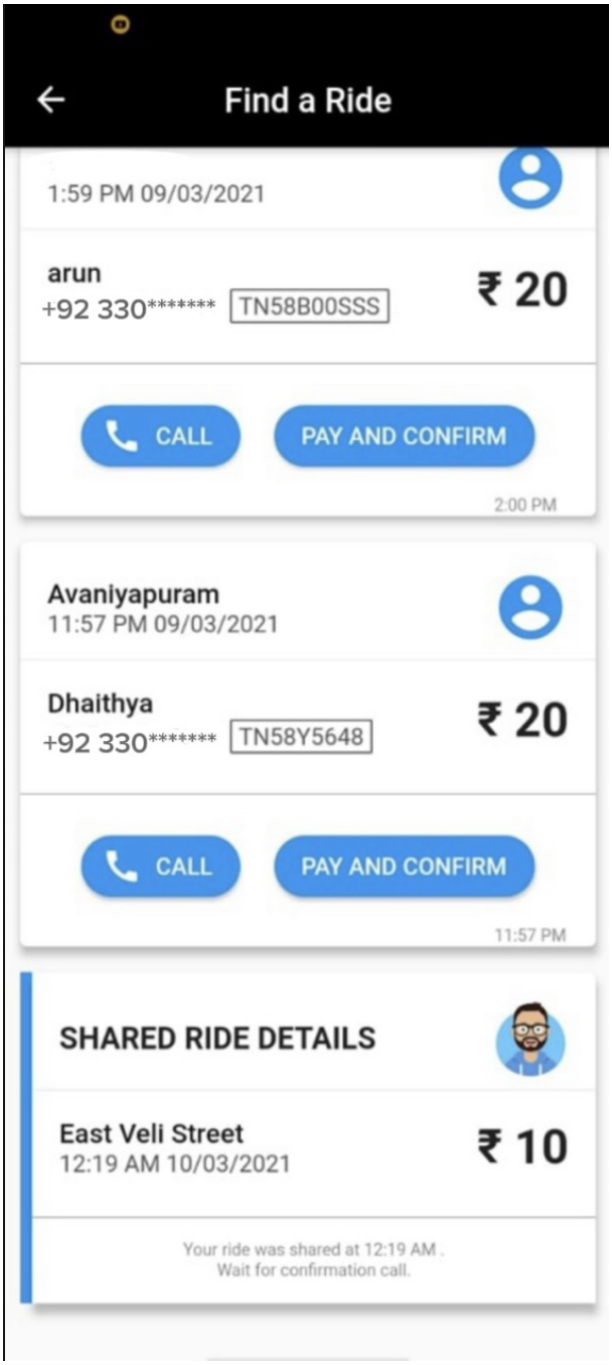

RideSync
localhost/CabBooking-PHP/admin/?page=clients
RideSync - Admin
Hajira Fatima
Dashboard
Manage Category
Cab Management
View Bookings
Registered Clients
System Users
Settings
List of Clients
Show 10 entries
Search:
Name Contact Status Action
1 Fatima, Hajira 123456789 hajirafatima110@gmail.com Active Action
2 Fatima, Hajira 123456790 hajirafatima1@gmail.com Active Action
3 Fatima, Hajira 123456790 hajirafatima123@gmail.com Active Action
4 Fatima, Hajira 1234567990 fatimahajira110@gmail.com Active Action
5 Fatima1, Hajira 1234567990 hajira110@gmail.com Active Action
6 Girls, SW 123456790 swgirls123@gmail.com Active Action
7 Girls, SW19 1234567901 Girlssoftware19@gmail.com Active Action
8 Shah, Zara 3456790123 zarashah@gmail.com Active Action
9 Software, Girls 123456790 swgirls@gmail.com Active Action
10 User, Testing 123456790 testinguser123@gmail.com Active Action
Showing 1 to 10 of 12 entries
Previous 1 2 Next
Type here to search
32°C Clear
8:39 AM
9/30/2023

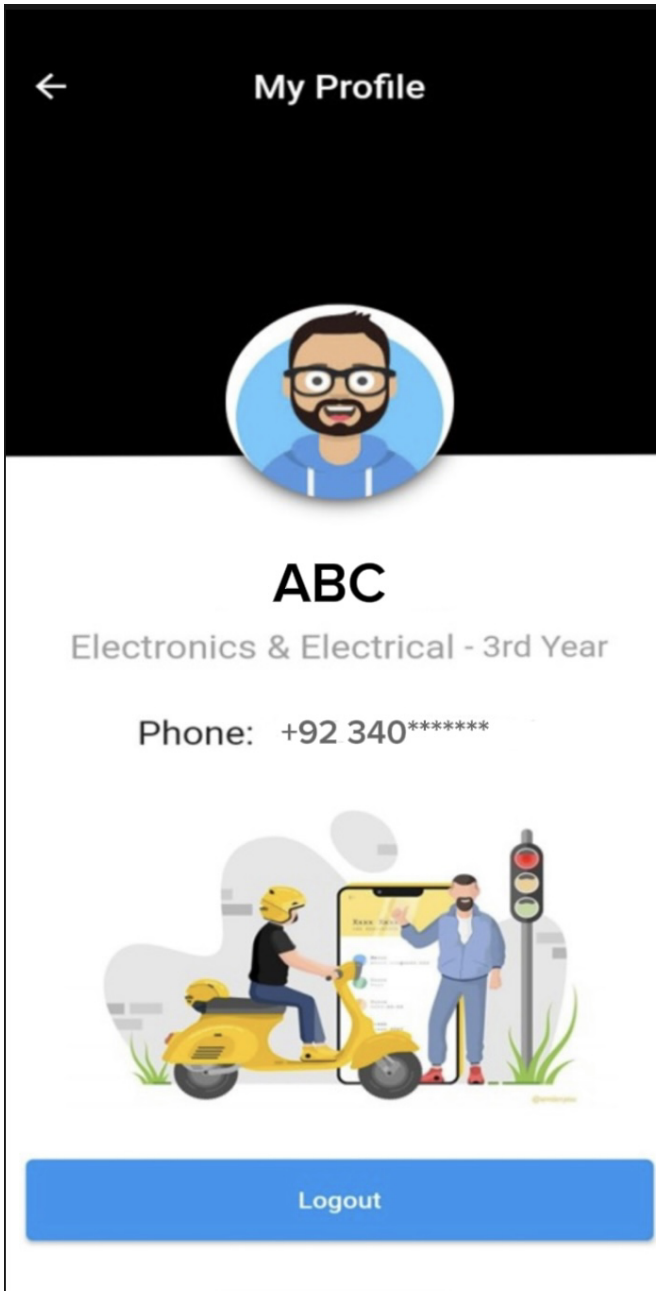

RideSync
localhost/CabBooking-PHP/admin/?page=user/list
RideSync - Admin
Hajira Fatima
Dashboard
Manage Category
Cab Management
View Bookings
Registered Clients
System Users
Settings
List of System Users
+ Add New User
Show 10 entries
Search:
Avatar Name Username Type Action
No data available in table
Showing 0 to 0 of 0 entries
Previous Next
© 2023. All rights reserved.
Developed By: Hajira Fatima
Type here to search
32°C Clear
8:39 AM
9/30/2023

MOBILE App


11:04
BOOK
RIDE
You'll receive a
4 digit code to verify next
+92 340*******
10/10
NEXT


11:34
Name
ABC
Branch: CSE
Year: 3
Do you have a vehicle to pool?
Yes:
No:
Register


Find a Ride
1:59 PM 09/03/2021
arun
+92 330*******
TN58B00SSS
₹ 20
CALL
PAY AND CONFIRM
2:00 PM
Avaniyapuram
11:57 PM 09/03/2021
Dhaithya
+92 330*******
TN58Y5648
₹ 20
CALL
PAY AND CONFIRM
11:57 PM
SHARED RIDE DETAILS
East Veli Street
12:19 AM 10/03/2021
₹ 10
Your ride was shared at 12:19 AM .
Wait for confirmation call.


My Profile
ABC
Electronics & Electrical - 3rd Year
Phone: +92 340*******
Logout

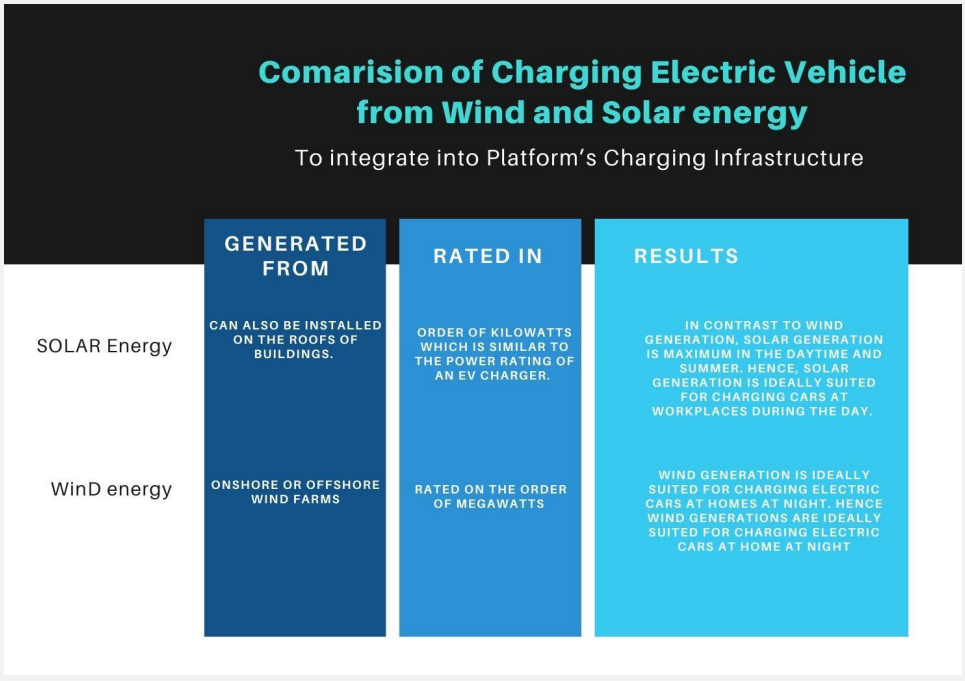
Conclusion

The electrifying synergy between EV charging stations and ride-sharing platforms holds great promise for the future of sustainable transportation. By strategically integrating these solutions, EV owners can conveniently access charging infrastructure while ride-sharing services benefit from improved station utilization. This symbiotic relationship contributes towards reducing range anxiety, promoting green mobility, and accelerating the adoption of electric vehicles. As technology advancements continue to streamline the convergence of these solutions, the future of electric vehicles and ride-sharing looks bright and sustainable.

Key Findings

Several charging infrastructure providers have successfully integrated renewable energy:

Tesla Supercharger Network: Tesla integrates solar panels at many of its charging stations, reducing grid dependence and showcasing the feasibility of renewables.
Enel X's e-Mobility Solutions: Enel X incorporates wind and solar power into its charging infrastructure, promoting eco-friendliness and grid stability.



wind generation is ideally suited for charging electric cars at homes at night. Hence wind generations are ideally suited for charging electric cars at home at night. In contrast to wind generation, solar generation is maximum in the daytime and summer. Hence, solar generation is ideally suited for charging cars at workplaces during the day.

Another solution to help match renewable generation and EV charging is smart charging. When solar and wind output is high, EV charging power can be increased and vice versa. This has the dual benefit of making EVs sustainable by using more green energy and reducing stress on the grid due to large-scale renewable energy generation.

References

https://e-vehicleinfo.com/charging-an-ev-using-renewable-energy-solar-wind/
https://electrek.co/2018/05/12/tesla-supercharger-2018-growth/
https://corporate.enelx.com/en/media/news/2022/04/enel-x-way-electric-mobility
https://chargedevs.com/newswire/volkswagen-and-enel-x-way-to-deploy-3000-high-power-charging-points-in-italy/

Appendices

Code Snippets

Main.dart

```
import 'package:flutter/material.dart';

import 'package:klndrive/HomeScreen/homeScreen.dart';

import 'package:klndrive/HomeScreen/findaRide.dart';
```

```
import 'package:klndrive/Profile/profile.dart';
import 'package:klndrive/auth/registerUser.dart';
import 'package:klndrive/auth/otpPage.dart';
import 'package:klndrive/auth/otpHome.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:shared_preferences/shared_preferences.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Firebase.initializeApp();
  SharedPreferences prefs = await SharedPreferences.getInstance();
  var userName = prefs.getBool("userName");
  runApp(MyApp(screen: (userName==null) ? OtpHome() : HomeScreen()));
}



class MyApp extends StatelessWidget {
  final Widget screen;
  MyApp({this.screen});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Car Booking System',
      //test phase
      home: screen,
      routes: {
        '/otphome' : (context) => OtpHome(),
        '/otppage' : (context) => OtpPage(),
        '/homescreen' : (context) => HomeScreen(),
        '/register'   : (context) => SignUp(),
```

```
      '/profile'    : (context) => ProfilePage(),
      '/findaride' : (context) => FindaRide(),
    },
  );
 }
}
```